



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# WEBOVÁ PREZENTACE PROSTOROVÝCH DAT

## Bakalářská práce

*Studijní program:* B2646 – Informační technologie  
*Studijní obor:* 1802R007 – Informační technologie  
*Autor práce:* **Tomáš Jodas**  
*Vedoucí práce:* Mgr. Kamil Nešetřil





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# SPATIAL DATA PRESENTATION ON THE WEB

## Bachelor thesis

*Study programme:* B2646 – Information Technology  
*Study branch:* 1802R007 – Information Technology  
*Author:* **Tomáš Jodas**  
*Supervisor:* Mgr. Kamil Nešetřil



Tento list nahradte  
originálem zadání.

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

## **Poděkování**

Rád bych poděkoval Mgr. Kamilu Nešetřilovi za jeho pomoc a ochotu během vedení bakalářské práce a pomoc při zpracování. Dále děkuji rodině za trpělivost a podporu při studiu na vysoké škole.

## **Abstrakt**

Cílem bakalářské práce je vytvoření nové webové aplikace pro zobrazení nasbíraných údajů v přehledných tabulkách a grafech. Část práce je věnována zhodnocení možností jednotlivých řešení v odvětví vizualizace geoprostorových dat. Hlavní část je věnována samotnému postupu vývoje aplikace, postupu zavedení vlastní mapy z geoportálu a také vytvoření datového modelu pro uchování informací. Dále je v práci uveden postup pro zpřístupnění aplikace z internetu.

### **Klíčová slova:**

WMS, WMTS, PostgreSQL, Geoportál, Webová aplikace, OpenLayers, jQuery, jqPlot

## **Abstract**

Purpose of this bachelor thesis is to create a new web application to display the collected data in tables and graphs. Part of the thesis is devoted to the evaluation of the capabilities of the individual solutions for the visualization of geospatial data. The main part is devoted to the process of application development, process implementation own maps of geoportal and create a data model for storing information. The thesis also shows you how to make the applications available on the internet.

### **Key words:**

WMS, WMTS, PostgreSQL, Geoportal, Web application, OpenLayers, jQuery, jqPlot

## Obsah

Prohlášení.....	4
Poděkování.....	5
Abstrakt.....	6
Obsah .....	7
Seznam ilustrací.....	9
Zkratky.....	10
1 Úvod.....	11
2 Problematika podrobněji.....	12
2.1 Existující řešení pro vizualizaci .....	12
2.1.1 Statsilk .....	13
2.1.2 InstantAtlas .....	14
2.2 Řešení mapových knihoven .....	15
2.2.1 OpenLayers.....	16
2.2.2 HSLayers .....	17
2.2.3 Google Maps API .....	18
2.2.4 LeafletJS .....	19
3 Cíle práce .....	20
4 Návrh řešení.....	22
4.1 Datový model .....	22
4.2 Server PostgreSQL.....	22
4.3 Rozvržení aplikace.....	22
4.4 Mapový podklad.....	23
4.4.1 WMS.....	23
4.4.2 WMTS .....	24
4.4.3 WFS .....	25
4.4.4 Umístění aplikace .....	25

5	Realizace řešení .....	26
5.1	Návrh databáze .....	26
5.1.1	Příklady vytvářecích skriptů .....	27
5.2	Databázový server .....	28
5.3	Použité technologie v aplikaci .....	29
5.3.1	jQuery .....	29
5.3.2	AJAX .....	29
5.3.3	jqPlot .....	29
5.4	Návrh vlastní aplikace .....	30
5.5	Hosting aplikace .....	37
6	Vyhodnocení řešení .....	38
6.1	Serverové požadavky .....	38
6.2	Podpora webových prohlížečů .....	38
6.3	Výběr databázového serveru .....	39
6.4	Výběr mapového podkladu .....	39
7	Shrnutí práce .....	40
	Citace .....	42



## Seznam ilustrací

Obrázek 1: Ukázka rozhraní Statsilk .....	13
Obrázek 2: Ukázka rozhraní InstantAtlas .....	14
Obrázek 3: Ukázka rozhraní OpenLayers .....	16
Obrázek 4: Ukázka rozhraní na Národním portálu INSPIRE .....	17
Obrázek 5: Ukázka rozhraní Google map .....	18
Obrázek 6: Ukázka rozhraní Leaflet s použitím OpenStreetMap .....	19
Obrázek 7: Reprezentace dlaždic při jednotlivých přiblížení .....	24
Obrázek 8: Výsledný datový model .....	26
Obrázek 9: Wireframe .....	30
Obrázek 10: Značka vrtu .....	33
Obrázek 11: Ukázka popup okna .....	34
Obrázek 12: Příklad zobrazení grafu .....	35
Obrázek 13: Konečný vzhled aplikace .....	36

## Zkratky

AJAX – Asynchronous JavaScript and XML

CSS – Cascading Style Sheet

FTP – File Transfer Protocol

PHP – starší Personal Home Page, nyní se používá PHP: Hypertextový Preprocesor

OGC – Open Geospatial Consortium

SDK – Software Development Kit

URL – Uniform Resource Locator

WFS – Web Feature Service

WMS – Web Map Service

WMTS – Web Map Tile Service

## 1 Úvod

V rámci informačního systému, který vzniká v rámci projektu MARE, jsou sledovány různé složky životního prostředí. Sledován je stav vody, včetně obsažených rozpuštěných látek, kterými jsou vápník, železo, hořčík, chloridy a podobně.

Existuje několik řešení, avšak některá komerční, či bez zdrojových kódů a pro budoucí rozvoj těžko rozšiřitelná, pro již existující data spíše nevhodná. Z tohoto důvodu je třeba navrhnout individuální řešení odpovídající požadavkům, splňující bezproblémové rozšíření a takovou aplikaci, která zobrazí požadované informace.

Výsledná aplikace může být nasazena na webhostingovém řešení, či na privátním webovém serveru s vlastním doménovým jménem pro pohodlný přístup.

## 2 Problematika podrobněji

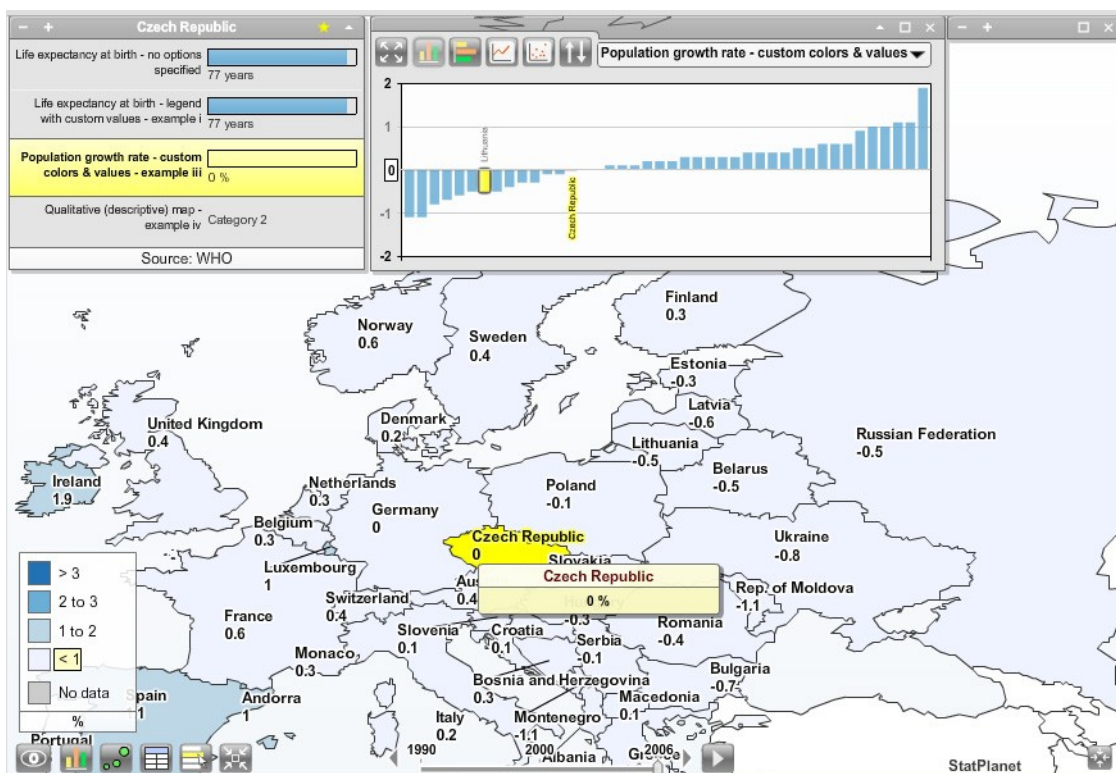
Ve své aplikaci vycházím částečně také ze svého bakalářského projektu v druhém ročníku [7], kde bylo třeba zobrazovat do jednoduché Google mapy stanice měřící průtok a teplotu z již vytvořené a naplněné MySQL databáze. Hodnoty pak byly zobrazeny do tabulky a grafu. Ověřil jsem si zde funkčnost některých komponent aplikace, nicméně pro novou aplikaci jsem potřeboval navrhnout zcela nový datový model a zobrazovat také jiná data. Samotný mapový podklad bude taktéž jiný a více zaměřený pro zobrazení více vrstev.

### 2.1 Existující řešení pro vizualizaci

Existuje mnoho vizualizačních nástrojů pro prostorová data. Většinou se jedná o nástroje pro sledování států a kontinentů. Pro bližší sledování na úrovni států či krajů, je třeba dalších úprav a přizpůsobení. To znamená další vynaložené finance. Existující nástroje nabízejí pouze data velkých států, pro Českou republiku by bylo nutné získat nebo vytvořit vlastní podklad. Některá řešení srovnám v následujícím souhrnu.

### 2.1.1 Statsilk

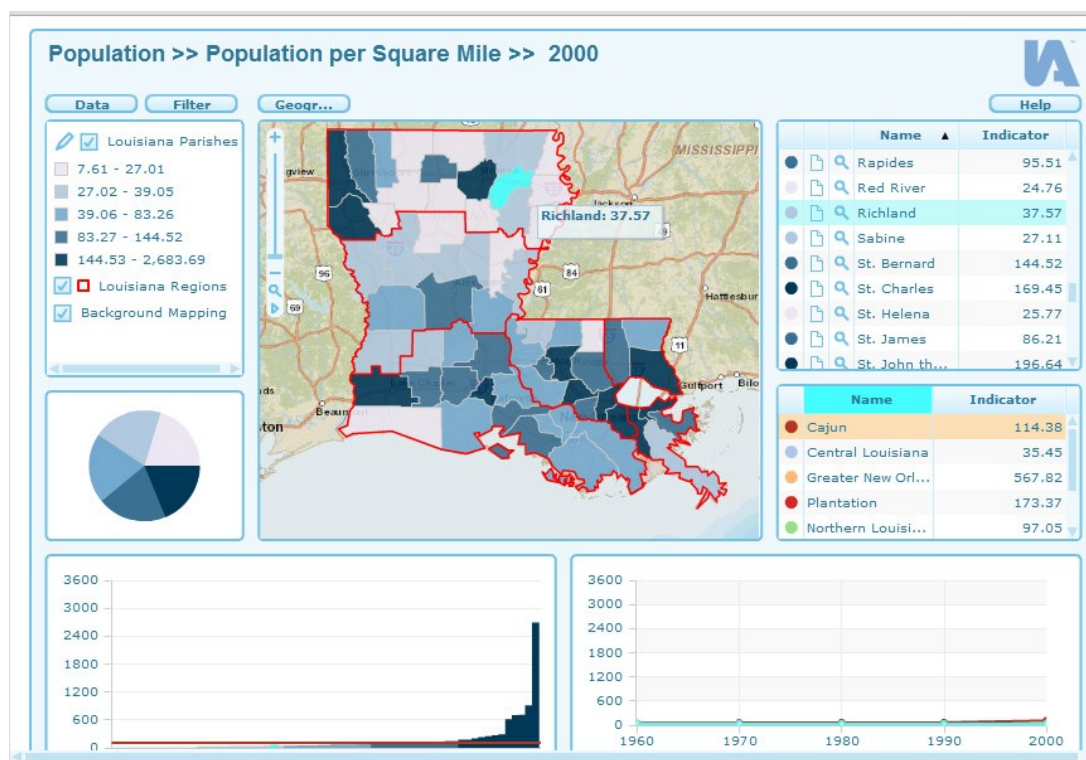
V případě systému ze stránek Statsilk [9] dostaneme komplexní řešení. Nedovoluje však jakoukoliv úpravu zdrojového kódu, celé řešení je flashovou aplikací a data jsou načítána ze souboru csv. Jednou z nepříznivých vlastností aplikace je nemožnost zobrazení detailní mapy České republiky. Je zde možnost načíst mapu pomocí ESRI shapefile.



Obrázek 1: Ukázka rozhraní Statsilk

### 2.1.2 InstantAtlas

InstantAtlas [8] je komerční aplikace, s možností výběru desktopového nebo serverového řešení. Aplikace InstantAtlas nabízí vytváření map, tabulek i grafů. InstantAtlas taktéž zobrazuje informace pomocí flashové aplikace. Ceny nabízených služeb za vytvoření požadované aplikace se pohybují od 770 dolarů nebo 595 euro.



Obrázek 2: Ukázka rozhraní InstantAtlas

## 2.2 Řešení mapových knihoven

V této oblasti existuje několik řešení, všechny fungují jako javascriptová knihovna a je tak možné s nimi dále pracovat. Všechny knihovny nabízejí v základním nastavení přibližně stejné funkce. Liší se především v rozšíření a také komunitou kolem každého z řešení. Z uvedeného následně vyplývají různě kvalitní příklady použití některých komponent.

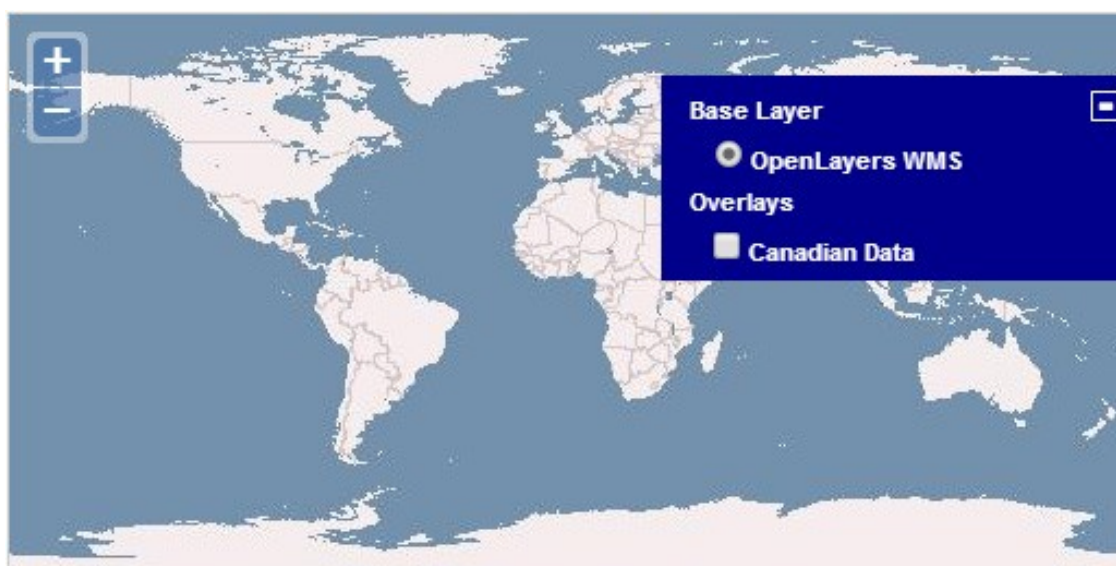
V tomto ohledu dle mého názoru vyniká OpenLayers. Komunita je široká a lze dohledat mnoho příkladů, řešení nejrůznějších problémů a rad.

### 2.2.1 OpenLayers

Knihovna je rozsáhlým projektem poskytovaný zdarma. Nabízí široké možnosti pro zobrazování map pomocí služeb WMS, WMTS, WFS a jiné. Umožňuje do mapy přidávat jednotlivé body nebo čáry. Umí také načíst Google mapy, od firmy Microsoft Mapy Bing nebo dokonce Mapy Seznam.

Současná verze OpenLayers [14] nabízí možnost načítání mapových podkladů, v České republice například z Národního geoportálu INSPIRE pomocí služby WMS a WMTS. Jedná se o projekt nadace Open Source Geospatial Foundation a je poskytován pod licencí FreeBSD.

Výhodou je aktuálnost projektu, poslední aktualizace je datována k březnu roku 2014. Do budoucna se počítá s novou verzí OpenLayers 3. Knihovna bude celá přepsána do současných webových technologií HTML5 a CSS3. Počítá se s podporou knihoven OpenGL, která by měla zvýšit výkon celého mapového portálu.



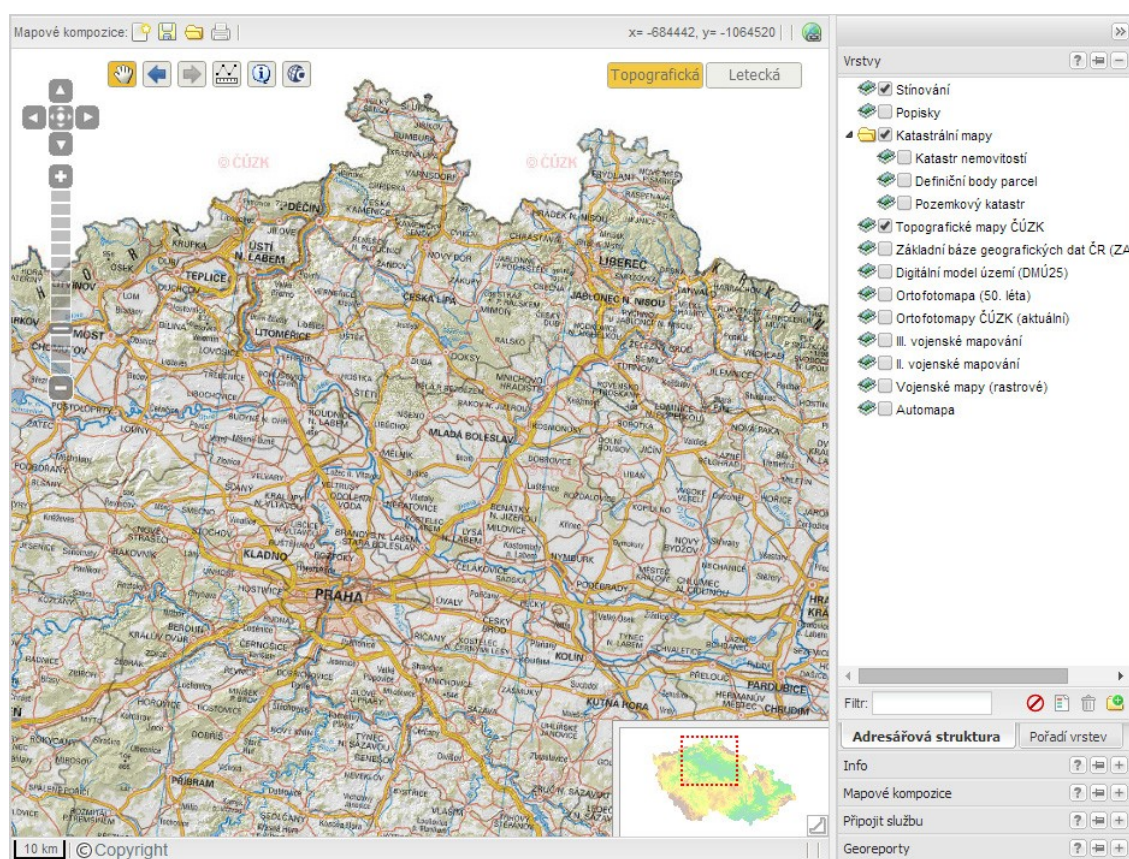
Obrázek 3: Ukázka rozhraní OpenLayers



## 2.2.2 HSLayers

Knihovna HSLayers [15] slouží k vytvoření mapového portálu. Funkčností vychází z OpenLayers – je postavena na kombinaci OpenLayers s rozšířením ExtJS [17] a umožňuje tedy rozšíření základních OpenLayers o nové funkce. Jedná se o Opensource projekt pod licencí GNU/GPL. Nevýhodou je neaktuálnost projektu, poslední aktualizace, dle stránek, proběhla v listopadu roku 2012.

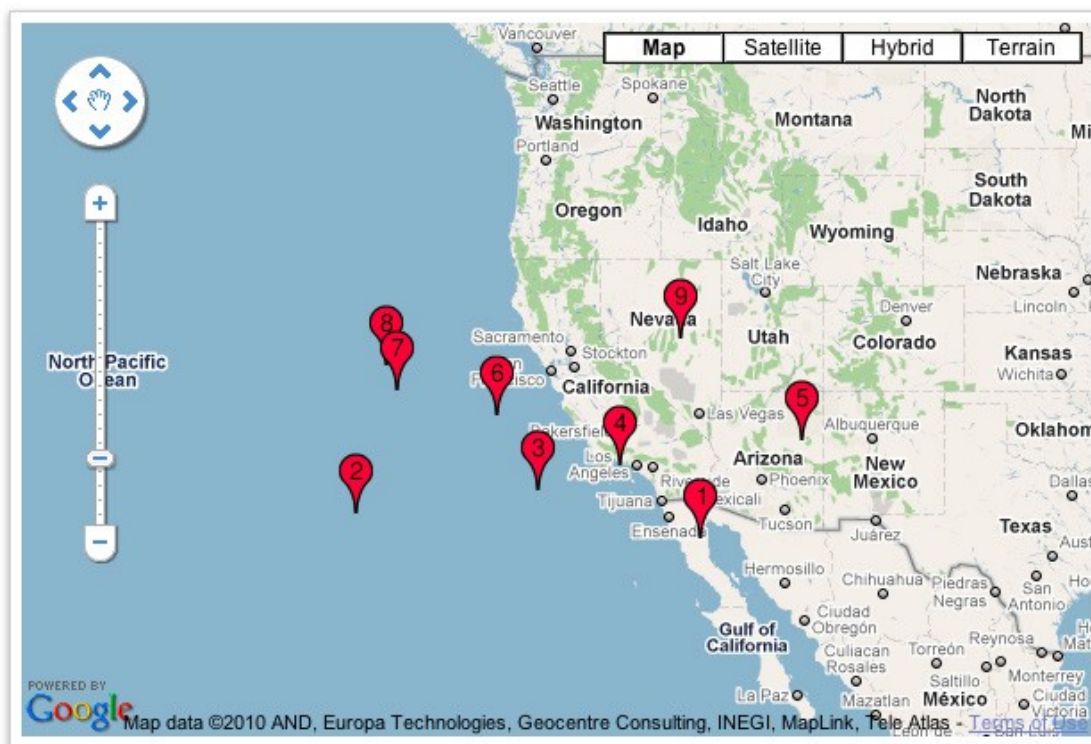
Díky rozšíření ExtJS je možné například zařadit jednotlivé vrstvy do skupin, což původní OpenLayers neumí.



Obrázek 4: Ukázka rozhraní na Národním portálu INSPIRE

### 2.2.3 Google Maps API

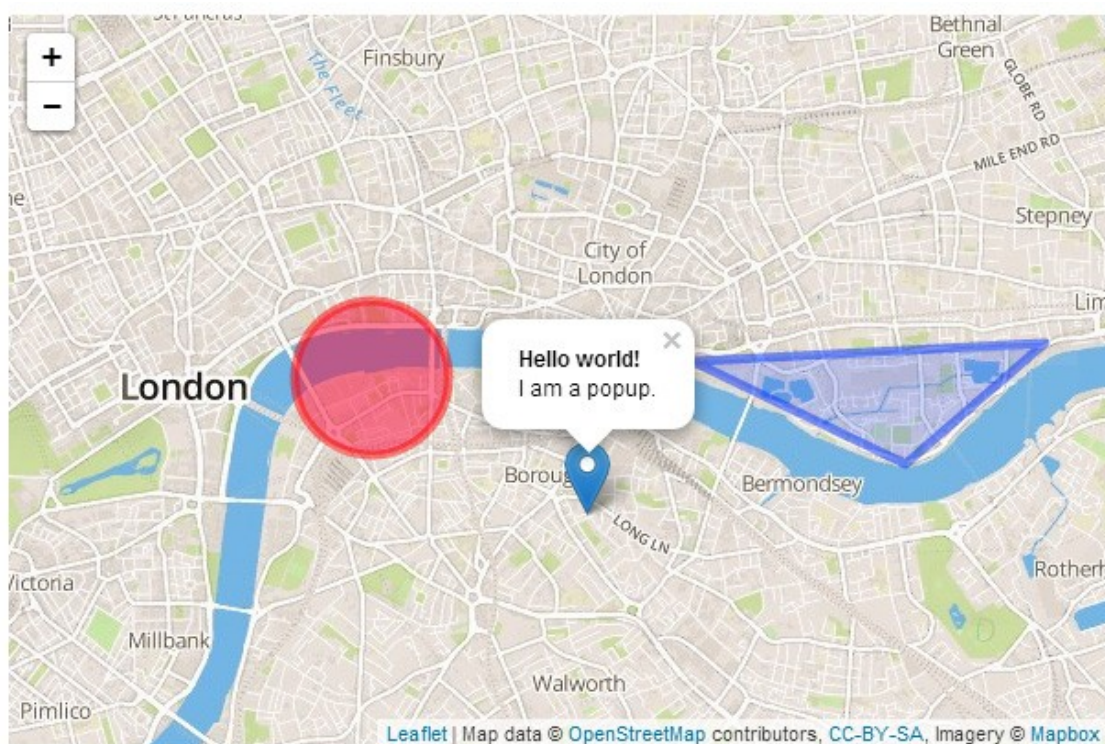
Mapové rozhraní firmy Google [18] je poskytováno zdarma. Pro rozsáhlejší projekty je nutná registrace pro vygenerování unikátního přístupového klíče. Pro základní zobrazení Google mapy registrace nutná není. Je zde také možnost načíst mapy z geoportálu pomocí služby WMS, avšak možnosti nejsou tolik rozsáhlé jako u OpenLayers. Nabízí také SDK pro vytváření aplikací pro mobilní telefony.



Obrázek 5: Ukázka rozhraní Google map

## 2.2.4 LeafletJS

Knihovna LeafletJS má nejmenší velikost v porovnání s ostatními. Záměrem je neposkytovat veškerou funkcionalitu v jednom balíku. Základní verze podporuje práci s vektory, umístění značek do mapy nebo práci s WMS vrstvami. Leaflet je také možné rozšiřovat pomocí zásuvných modulů třetích stran a získat tak další funkcionalitu.



Obrázek 6: Ukázka rozhraní Leaflet s použitím OpenStreetMap



### 3 Cíle práce

Jedním z cílů je seznámení se se současným datovým modelem EI MARE v aplikaci Microsoft Access a dle stávajícího řešení nastínit nový, normalizovaný model. Je třeba nalézt aplikaci pro datové modelování, ve které lze vhodným způsobem definovat jednotlivé tabulky a jejich propojení pro uchování dat. Běžnou výbavou takového programu je možnost exportu vytvořeného modelu do formátu vybraného databázového systému. Vhodné navržení usnadní manipulaci s daty při programování a ve výsledku také jejich zobrazení. Pro výše popsany cíl je potřebné mít již zprovozněný databázový server a výsledný vyexportovaný skript zde odladit s testovacími daty. Data mi následně budou nahrána vedoucím práce, naplnění databáze tak není předmětem mé práce.

Hlavním cílem je vytvořit webovou aplikaci pro sledování měřených veličin ve vrtech v okolí měst Ústí nad Labem a Most v částečně zaplavených rekultivovaných dolech. Aplikace bude načítat data z vytvořeného modelu. Z databáze si načtu jednotlivé vrty, které budou uživateli zobrazeny nad mapovým podkladem. Jednotlivé vrty budou zobrazeny jako barevně odlišené body. Barevností se budou lišit dle svého druhu a účelu pro zajištění lepší přehlednosti. Úmyslem je také umožnit návštěvníkovi vybírat mezi dvěma mapami dle možností mapového portálu. Předpokladem je však, že základní vrstvou bude rastrová mapa České republiky, jako volitelná vrstva pravděpodobně mapa zabývající se geomorfologickým rozložením. Údaje tak budou přehledně zobrazeny návštěvníkovi aplikace.

Výsledná mapa by měla být zaměřena na oblast zájmu s přiměřeným přiblížením. Úroveň se odvíjí od počtu a umístění jednotlivých stanic. Návštěvníkovi však umožním dále přibližovat, oddalovat a mapu posouvat dle libosti a zaměřit se tak podrobněji na menší oblast zájmu. Existuje několik knihoven poskytujících práci s mapami. Pro tento krok je zde několik možností – mimo jiné Google API pro práci s mapami nebo javascriptové OpenLayers, HSLayers či Leaflet. V tomto kroku se tedy zaměřím na jejich vhodnou volbu.

Jak bylo zmíněno, databázový server bude muset být zprovozněn jako první za účelem odladení datového modelu. Nicméně souběžně s tímto cílem lze také řešit umístění pro aplikaci. V tomto dalším kroku tedy bude cílem vybrat variantu webového hostingu, zdarma či placeného, nebo dokonce porovnat variantu robustního serverového řešení. Umístění databáze a samotné aplikace nemusí sdílet jeden server či webhostingovou

variantu. Části lze oddělit s podmínkou, že k serveru s databází bude existovat vzdálený přístup pro manipulaci s daty.

V neposlední řadě je cílem práce také záměr uživatelského komfortu při procházení aplikací. Není žádoucí s každým kliknutím myši znovu načítat kompletní stránku. Účelem je aktualizovat pouze vybrané části stránky. Funkčnost lze zajistit pomocí vhodně vybrané a naimplementované javascriptové knihovny.

## 4 Návrh řešení

### 4.1 Datový model

Po prostudování datového modelu EI MARE v programu Microsoft Access, jsem navrhl databázovou strukturu v programu SQL Power Architect [3]. Práce spočívá v optimálním navržení všech potřebných tabulek a vhodném vzájemném propojení. V mém zájmu je učinit databázi přehlednou i pro budoucí rozvoj.

Následně potřebuji z vytvořených tabulek získat skript pro jejich vytvoření na databázovém serveru PostgreSQL.

### 4.2 Server PostgreSQL

Vybraným databázovým serverem se stal PostgreSQL. Použiji poslední aktuální verzi. Administraci zajistím, s příslušnými právy k přístupu, webovou aplikací s názvem phpPgAdmin [19]. Obsluha je jednoduchá, zobrazení tabulek je přehledné a poskytne tak okamžitý souhrn obsažených informací. Rozhraní také dovoluje spouštět SQL skripty, mohu tedy pracovat přímo s určitými částmi databáze. Spouštěč SQL kódu použiji k vytvoření tabulek, vyexportovaných z programu SQL Power Architect.

Umožním také vzdálený přístup pomocí aplikace z klientského počítače. Díky programu následně obdržím data, která mi budou poskytnuta k budoucímu vývoji mé aplikace. Naplnění databáze není předmětem této práce.

Pro další postup ve vývoji bude potřeba již zprovozněný server s řádně otestovanými přístupy z prostředí klientské aplikace a webového rozhraní v phpPgAdminu.

### 4.3 Rozvržení aplikace

K úspěšnému navržení aplikace mi pomůže vytvoření schématu webové stránky neboli wireframe. Plánuji v celých stránkách mít dominantní část mapového podkladu, kde zobrazím jednotlivé stanice se základními informacemi. Informace bych zobrazil jako popup okno při přejetí myši.

Levý sloupec věnuji ovládacím prvkům. V horní části zobrazím výběr jednotlivých vrstev a všechny typy stanic pro snadné filtrování. Kliknutím myši se zobrazí příslušný výběr vrstvy. V druhé třetině nechám uživateli možnost výběru časového období a požadovaných sledovaných veličin s názvem constituents.

Horní část ponechám k zobrazení výstupního grafu. Na ose X zobrazím vždy datum, na ose Y vyobrazím informace vztahující se k vybrané veličině. Dále ke každé veličině nechám do grafu zobrazit i její standardní hodnotu pro porovnání. V databázi jsou hodnoty pro Nařízení vlády č. 23/2011 Sb., příloha č. 3, tabulka 1a. Při pohybu myši v grafu by se mělo zobrazovat popup okno s údaji o hodnotách. K vykreslení použiji javascriptovou knihovnu jqPlot, která je poskytována zdarma a její možnosti odpovídají požadavkům pro vytvářenou aplikaci.

#### 4.4 Mapový podklad

Základ podkladu bude javascriptová knihovna OpenLayers, umožňující různé možnosti pro jednotlivé mapové vrstvy. Bohaté možnosti nabízí také v zobrazení jednotlivých bodů do vytvořené vrstvy s přiřazením potřebných údajů z databáze, kterými jsou souřadnice bodu, jednoznačná odlišitelnost pomocí unikátního id čísla a podobně.

Budou využity mapy z Národního geoportálu INSPIRE [1]. Portál pracuje pouze s několika mapovými službami pro zobrazení. Konkrétně se jedná o WMS, WMTS a službu pro ArcGIS server. Pro své účely využiji první dvě jmenované, spolu s WFS a dalšími službami jsou to standardy Open Geospace Consorcia (OGC) [5].

##### 4.4.1 WMS

Služba [10] nabízí přístup pomocí HTTP rozhraní a to pomocí dvou metod *GET* a *POST*. Požadavek na WMS definuje geografickou vrstvu oblasti zájmu. Jako odpověď poskytuje rastrové mapové podklady ve formátech JPEG, PNG a jiné. Poslední aktuální verze 1.3.0 je také standardem ISO s označením ISO 19128.

Operace definované pro WMS se nazývají GetCapabilities, GetMap, a GetFeatureInfo:

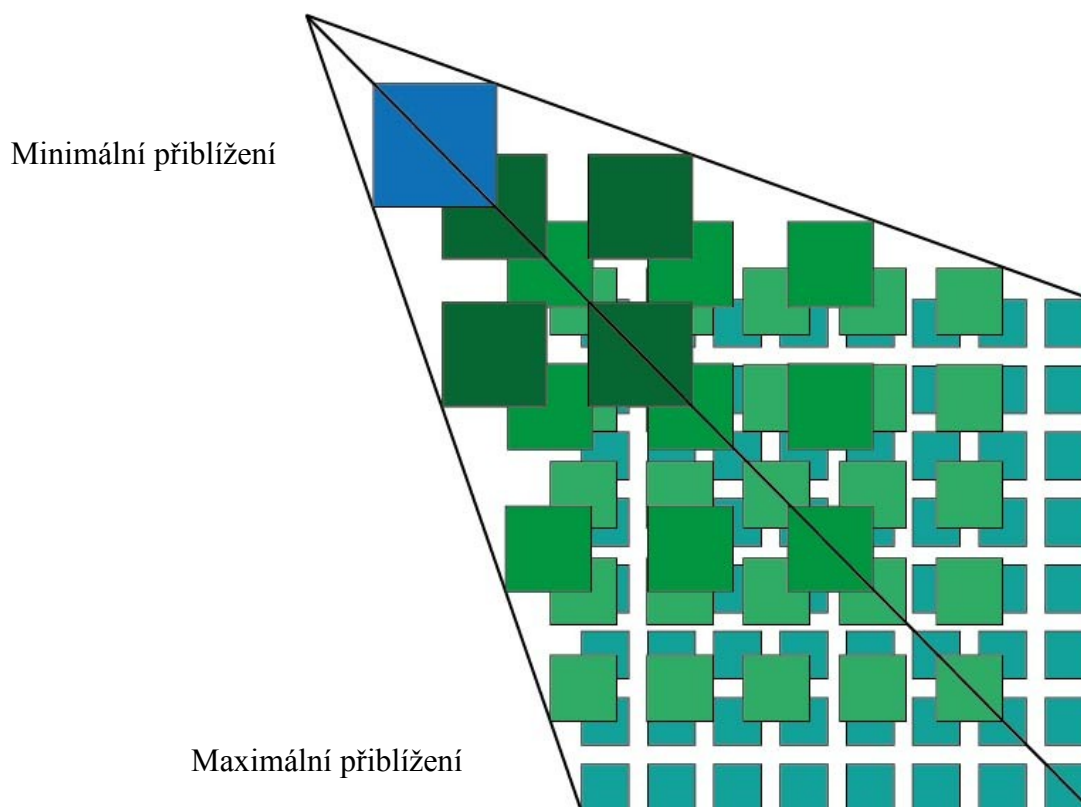
- GetCapabilities – účelem operace je získat informace o metadatech, strojově čitelný popis obsahu serveru; jedná se o povinnou položku
- GetMap – operace vydává mapu dle žádosti, v opačném případě vyvolá výjimku; jedná se o povinnou položku
- GetFeatureInfo – jedná se o nepovinnou operaci; účelem je získat podrobnější informace, například souřadnice k vybranému bodu

#### 4.4.2 WMTS

WMTS služba [11] poskytuje oproti WMS již předrenderované dlaždice v daném měřítku mapy (Obrázek 7: Reprezentace dlaždic při jednotlivých přiblížení). OGC ho poprvé zveřejnilo roku 2010. Oproti předchozí službě jsou zde poskytovány již připravené mapové dlaždice. Požadavkem na server je, aby vrátil dlaždice co nejrychleji. Vývoj služby probíhá samostatně od WMS.

Stejně jako WMS, definuje WMTS také tři operace GetCapabilities, GetTile, a GetFeatureInfo

- GetCapabilities – operace je stejná jako u WMS, tedy získat informace o metadatech; jedná se o povinnou položku
- GetTile – dle žádosti operace poskytne konkrétní mapovou dlaždici; jedná se o povinnou položku
- GetFeatureInfo – jedná se o nepovinnou operaci; účelem je získat podrobnější informace, například souřadnice k vybranému bodu



Obrázek 7: Reprezentace dlaždic při jednotlivých přiblížení



#### 4.4.3 WFS

WFS [12] je vektorovou službou na rozdíl od WMS, poskytující pouze rastrové obrazy. Umožňuje načítat data například ve formátu GML [6], která skutečně potřebujeme oproti načítání celých zdrojových souborů. WFS ve své poslední verzi je také ISO standardem s označením ISO 19142.

#### 4.4.4 Umístění aplikace

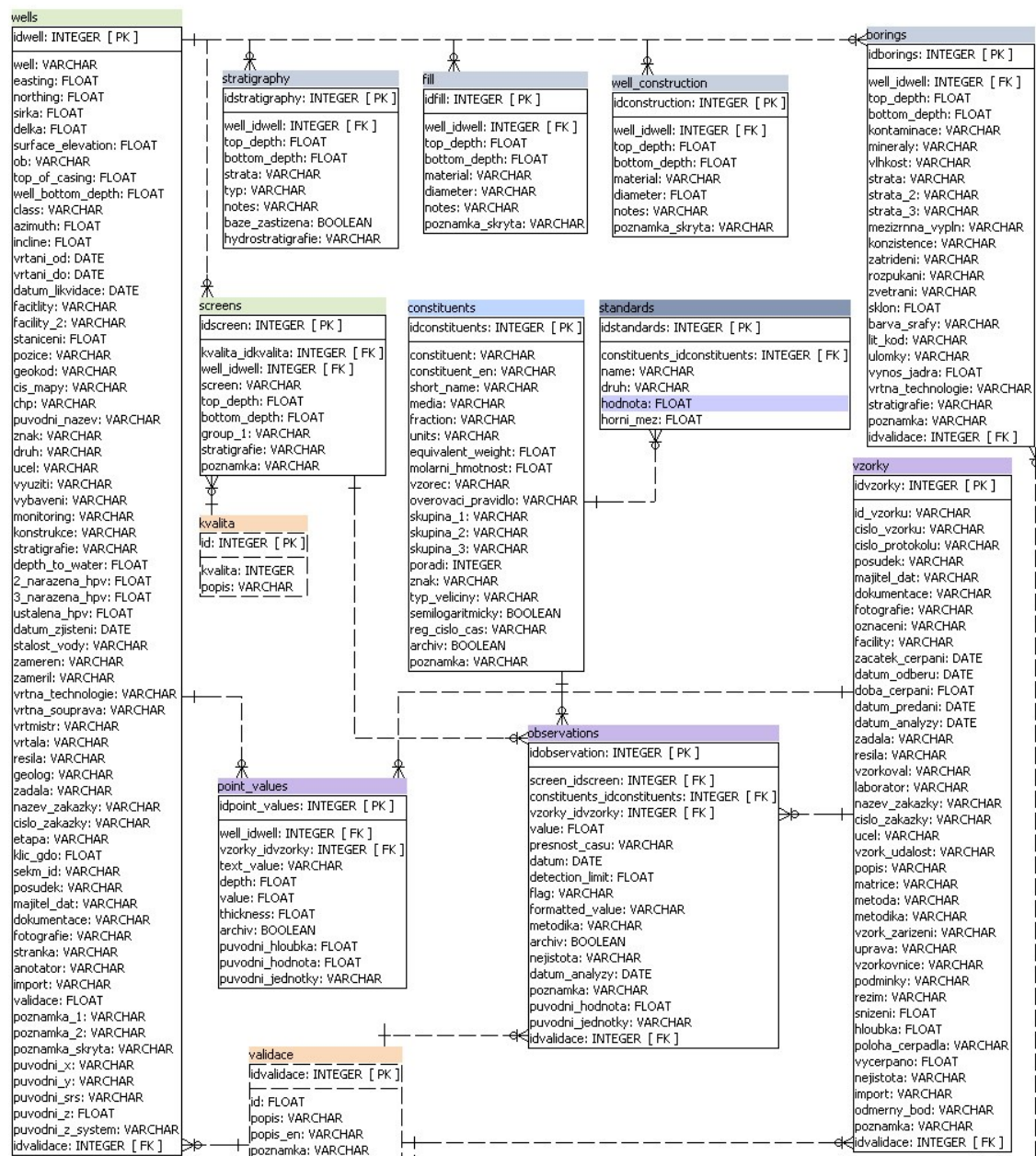
Samotná aplikace nemá výrazné požadavky. Provoz vyžaduje pouze webový server s podporou PHP skriptů a databázový server PostgreSQL. Mé rozhodnutí směřuje k nasazení na svém virtuálním serveru, kde provozuji také své další webové projekty. Je dostatečně výkonný pro celé řešení a mohu si nakonfigurovat přístup pro obsluhu, jak budu potřebovat. Nejsem omezen žádným parametrem, zejména pamětí, výpočetním výkonem a diskovým prostorem.

Celý projekt bude také přístupný z internetu. Neomezený provoz serveru zaručuje jeho bezproblémovou dostupnost.

## 5 Realizace řešení

### 5.1 Návrh databáze

Vytvořil jsem datový model dle existující databáze v Microsoft Access. Použiji všechny potřebné tabulky a převodu v modelovacím programu pro návrh databází SQL Power Architect. Jako databázový server volím PostgreSQL ve verzi 9.2 a výsledný skript pro vytvoření tabulek zde aplikuji.



Obrázek 8: Výsledný datový model

### 5.1.1 Příklady vytvářecích skriptů

Kód pro vytvoření tabulky pro stanice (wells), k ní se bude vztahovat další s názvem screens. Jejich vztah je 1:N, to znamená pro různé položky ve screens, odpovídá jedna položka ve wells. V druhé tabulce je sloupec s názvem well\_idwell, kam se ukládá číslo id z tabulky pro jednotlivé stanice. Obdobně funguje také třetí tabulka s názvem observations, kde uchovávám již konkrétní naměřené hodnoty. Tím je zaručen správný výběr položek.

**Ukázku jsem zkrátil o nesouvisející položky.**

```
CREATE TABLE mare.wells (  
    idwell INTEGER NOT NULL DEFAULT nextval(mare.wells_idwell_seq),  
    well VARCHAR NULL,  
    easting REAL NULL,  
    northing REAL NULL,  
    sirka REAL NULL,  
    delka REAL NULL,  
    druh VARCHAR NULL,  
    ucel VARCHAR NULL,  
    vyuziti VARCHAR NULL,  
    monitoring VARCHAR NULL,  
    CONSTRAINT idwell PRIMARY KEY (idwell)  
);
```

```
CREATE TABLE mare.screens (  
    idscreen INTEGER NOT NULL DEFAULT  
        nextval(mare.screens_idscreen_seq),  
    kvalita_idkvalita INTEGER NULL,  
    well_idwell INTEGER NULL,  
    top_depth REAL NULL,  
    bottom_depth REAL NULL,  
    poznamka VARCHAR NULL,  
    CONSTRAINT idscreen PRIMARY KEY (idscreen)  
);
```

```
CREATE TABLE mare.observations (  
    idobservation INTEGER NOT NULL DEFAULT  
        nextval('mare.observations_idobservation_seq'),  
    screen_idscreen INTEGER NULL,  
    value REAL NULL,  
    datum DATE NULL,  
    formatted_value VARCHAR NULL,  
    poznamka VARCHAR NULL,  
    puvodni_hodnota REAL NULL,  
    puvodni_jednotky VARCHAR NULL,  
    CONSTRAINT idobservation PRIMARY KEY (idobservation)  
);
```

## 5.2 Databázový server

Databázovým serverem se stal virtuální server u společnosti Wedos, kde již několik let provozují své soukromé projekty. Pro účel bakalářské práce se hodil jak výkonem, tak dostatečnou kapacitou disků a operační paměti. Virtuální server zajistí také webový prostor.

Počítač je vybaven dvěma gigabyty operační paměti a třiceti gigabyty diskového prostoru. Nainstalovaným operačním systémem je zde linuxová distribuce CentOS. Díky balíčkovacímu systému yum snadno doinstaluji databázový server PostgreSQL ve verzi 9.2.

Stažením a instalací potřebných balíčků, včetně souvisejících pro spolehlivý chod, pokračuji vytvořením uživatele a přidělím práva k databázi, sloužící k provozu webových stránek. Zprovozňuji také rozhraní phpPgAdmin, které využiji k budoucí obsluze databáze, ale hlavně k prvotnímu nahrání vytvořené struktury tabulek. Administrace je zabezpečena jménem a heslem, použiji již vytvořené údaje během instalace.

Po přihlášení vykonám SQL dotaz se svou databázovou strukturou a vytvořím tak potřebné tabulky. Aby byly naplněny informacemi, musel jsem umožnit přístup ze vzdáleného počítače. Na firewallu tedy povoluji potřebný TCP port 5432. Následně jsem čekal na vyplnění struktury finálními daty. Během vývoje se vyskytly drobné úpravy struktury, vliv na funkčnost však neměly. Server také každý týden pravidelně zálohuji, mnou vytvořeným skriptem, na vzdálený počítač prostřednictvím protokolu FTP.

## 5.3 Použité technologie v aplikaci

Aplikace je stavěna na značkovacím jazyku HTML a skriptovacím jazyku PHP. Uživatelský komfort zajišťuje javascriptová technologie AJAX s knihovnou jQuery, aby nedocházelo k opětovnému načítání stránky při uživatelské interakci. Grafy jsou zobrazovány pomocí stejné knihovny s rozšířením jqPlot.

### 5.3.1 jQuery

Knihovna jQuery zjednodušuje práci mezi javascriptem a HTML. První verze byla vydána roku 2006 a v dnešní době podporuje všechny hlavní webové prohlížeče. Umožňuje při výběru prvku ve stránce změnit jeho vlastnosti nebo reagovat na další elementy. Používá se také pro jednoduché animace, pro animované rozbalování prvků a formulářů. Jeho vlastností se také využívá ve formuláři ke kontrole zadaných povinných údajů a ověření duplicit při registraci spolu s technologií AJAX.

#### Ukázka kódu:

```
//Ve stránce se hledá element SELECT s id select_well_all a při změně
//provede naprogramovanou akci
//(možno spustit například ajaxovou událost)
$("#select#select_well_all").change(function(event) {
});
```

### 5.3.2 AJAX

Javascriptová technologie AJAX umožňuje návštěvníkovi stránek načítání dat z databáze nebo změnu obsahu stránky bez jejího opětovného načtení. Celé asynchronní načítání probíhá pomocí knihovny psané ve skriptovacím jazyce JavaScript.

#### Ukázka kódu:

```
//Odeslání dat metodou GET souboru get.php, který požadavek zpracuje.
$.ajax({
    type: 'get',
    url: 'get.php',
    data: 'id_stanice='+id_stanice+',
    success: function(data) {
        $('#left_panel_mid').html(data);
        //do části stránky identifikované
        //jako left_panel_mid se data odešlou
    }
});
```

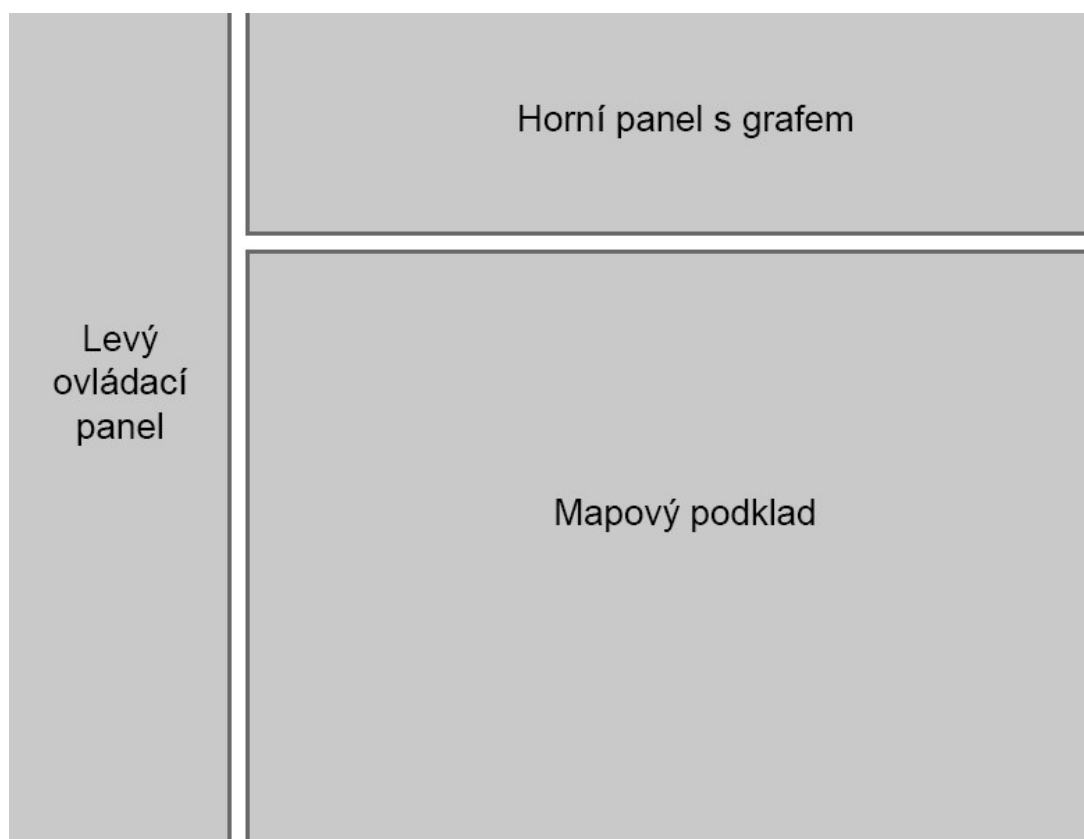
### 5.3.3 jqPlot

Rozšíření pro jQuery umožňuje vykreslovat grafy kruhové, sloupcové, spojnicové s různými nastaveními. V aplikaci využívám spojnicových grafů k zobrazování hodnot

v závislosti na datu jejich pořízení. Je možné z databáze načíst i další údaje o názvu veličiny včetně její příslušné jednotky.

## 5.4 Návrh vlastní aplikace

K dalšímu vývoji si potřebuji navrhnout základní umístění jednotlivých částí stránky takzvaný wireframe. V dalším kroku návrh převedu již na šablonu tvořenou HTML značkami se styly CSS.



Obrázek 9: Wireframe

Prvním krokem v návrhu aplikace bylo vytvoření mapového podkladu. Základem je OpenLayers umožňující pracovat s jakoukoliv dostupnou mapovou službou. V České republice je to například Národní geoportál INSPIRE. Základní vrstva je tvořena rastrovou topografickou mapou a jednotlivé mapové dlaždice vyvolávám pomocí služby WMTS. Mapa je při navštívení stránek dominantní v celé aplikaci. Další přidanou mapou pomocí služby WMS je geomorfologická mapa České republiky.

V celém skriptu mám nadefinováno základní nastavení OpenLayers s proměnnou map. Ponechal jsem zobrazení měřítka v levém dolním rohu mapy, ukazatel přiblížení

v horním rohu mapy. Menu s výběrem vrstev a prvků se vyskytuje mimo mapu v levém horním rohu.

Následně mé počínání začalo nastavením již konkrétní vrstvy služby WMTS. Nastavením jednoho z hlavních parametrů adresy, odkud se mapa bude načítat je URL: *[http://geoportal.gov.cz/arcgis/rest/services/CENIA/cenia\\_rt\\_RET/MapServer/WMTS](http://geoportal.gov.cz/arcgis/rest/services/CENIA/cenia_rt_RET/MapServer/WMTS)* Tím nezískám již funkční vrstvu, dále je nutné definovat jaké jednotlivé sety dlaždic pro jednotlivá přiblížení parametrem `matrixIds` načítat. Parametrem `isBaseLayer` nastavením na hodnotu `TRUE` učiním svou novou vrstvu jako základní a vždy bude automaticky vybrána při nové návštěvě stránek. Změnou na `FALSE`, se podklad vůbec nezobrazí, proto je lepší vůbec tento parametr nenastavovat v případě pouze jedné vrstvy. Ovšem při přidávání další mapové vrstvy musí být podklady jednoznačně odděleny a vybrána ta správná.

Posledním nastavitelným parametrem `visibility` na hodnotu `TRUE` určím viditelnost mapy po nahrání stránky. Vrstvu poté přiřadím proměnné `map` příkazem `addLayers()`. Výsledek příkaz vypadá následovně: `map.addLayer(wmts_vrstva)`. Budu-li požadovat přidání více vrstev najednou, vytvořím pole názvů a obdobným příkazem `addLayers([])` vložím. Celý příkaz vypadá `map.addLayers([vrstva_1, vrstva_2])`.

### Ukázka práce s WMTS

```
var wmts = new OpenLayers.Layer.WMTS({
  name: "Česká republika",
  url: "http://geoportal.gov.cz/arcgis/rest/
        services/CENIA/cenia_rt_RET/MapServer/WMTS",
  matrixSet: "EPSG:102067",
  matrixIds: [
    //zde je pole dlaždic pro jednotlivá přiblížení (zkráceno)
    {
      identifier: '0',
      tileWidth: 256,
      tileHeight: 256,
      scaleDenominator: 7741440
    }, {...}
  ],
  format: "image/jpeg",
  opacity: 1,
  isBaseLayer: true
});
```

## Popis kódu

Do proměnné `wmts` přiřadíme novou vrstvu WMTS pomocí `OpenLayers`. Potřebuji také nastavit příslušný název, který pak chci zobrazovat. Definuji také adresu mapového serveru, příslušnou projekci map, jednotlivé sady dlaždic pro různá přiblížení a tuto vrstvu mám nastavenou parametrem `isBaseLayer` jako hlavní.

Druhou přidanou vrstvou se stala geomorfologická mapa. Česká republika se skládá ze čtyř geomorfologických provincií, dělí se na subprovincie, dále na geomorfologické oblasti a poté geomorfologické celky. K dispozici mám tentokrát podporu WMS služby. Adresa požadavku na mapový server je: [http://geoportal.gov.cz/arcgis/services/CENIA/cenia\\_geolog\\_geomorf/MapServer/WMS/Server](http://geoportal.gov.cz/arcgis/services/CENIA/cenia_geolog_geomorf/MapServer/WMS/Server).

Ke správnému fungování potřebuji vybrat také jednotlivé vrstvy zmíněných provincií, oblastí a celků. Následujícím nastavením parametru `isBaseLayer` na `TRUE` a parametru `visibility` na `FALSE` mi způsobí možnost výběru mezi touto a WMTS vrstvou. Výsledná přidaná mapa všechno dělení obsahuje, my tak dostaneme kompletní vrstvu. Příkazem `map.addLayers(wms_vrstva)` opět přidávám proměnné `map` novou mapu. Po aktualizaci stránek mohu nyní vybírat mezi vrstvami, zobrazit si jakou zrovna potřebuji a hodí se pro mé potřeby.

## Ukázka práce s WMS

```
var geo = new OpenLayers.Layer.WMS(  
    "Geomorfologická mapa",  
    "http://geoportal.gov.cz/arcgis/services/CENIA/cenia_geolog_geomorf/MapServer/WMS/Server",  
    {layers: "0,1,2,3,4,5,6,7",  
      srs:"EPSG:102067",  
      format: "image/jpeg"},  
    { isBaseLayer: true, visibility: false, projection: new  
      OpenLayers.Projection('EPSG:102067') }  
);
```

## Popis kódu

Na prvním řádku je proměnná `geo`, které sdělíme, že budeme pracovat s `OpenLayers` a službou WMS. Dále vyplníme nutné položky. Název vrstvy (Geomorfologická mapa), adresu portálu odkud se načítá mapa, vybrané vrstvy. Zde je také možné definovat projekci map ve správném souřadnicovém systému, jako podporuje mapový server.



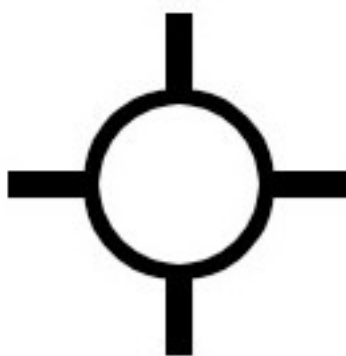
V mém případě to nutné není, projekci jsem již definoval u první vrstvy pomocí WMTS.

Nyní přišla řada na práci s databází. V PHP skriptu tedy vytvořím příkazem `pg_connect()` připojení k databázi s příslušnými přihlašovacími údaji.

### **Ukázka kódu s připojením k PostgreSQL databázi**

```
pg_connect("host=IP_Adresa_serveru  
           dbname=nazev_databaze  
           user=uzivatel  
           password=heslo");
```

Do mapy poté umístí jednotlivé stanice, každá její ikona je barevně odlišena dle jeho druhu a účelu. Zde je dvojí možnost umístění bodů do mapy. První, jednodušší způsob, je zobrazit značku do mapy jako obyčejný bod. Tím je ztracena detailnější práce s objektem, není možné objektu přiřadit podrobnější informace z databáze.

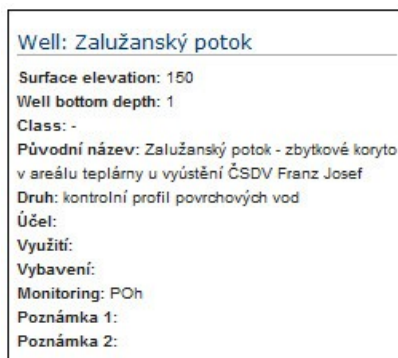


**Obrázek 10: Značka vrtu**

Existuje tedy i druhý způsob zobrazení a tím je vyobrazit objekt jako vektor. K němu je možné přiřadit jakékoliv databázové informace. Údaje využívám při najetí myši přes vrt v mapě. Zobrazí se malé popup okno s detailním výpisem důležitých položek náležících vrtu, pokud jsou v databázi vyplněné. Okno je poté skryto, co kurzor myši opustí danou pozici.

Nejprve si načtu z databáze všechny unikátní druhy a účely stanic. Přiřadím barevnou ikonu, abych zajistil, že každý druh a účel bude mít vlastní barvu. Vytvořená ikona vychází ze značky vrtu (Obrázek 10: Značka vrtu), barevně se mění pouze její kulatý bílý střed.

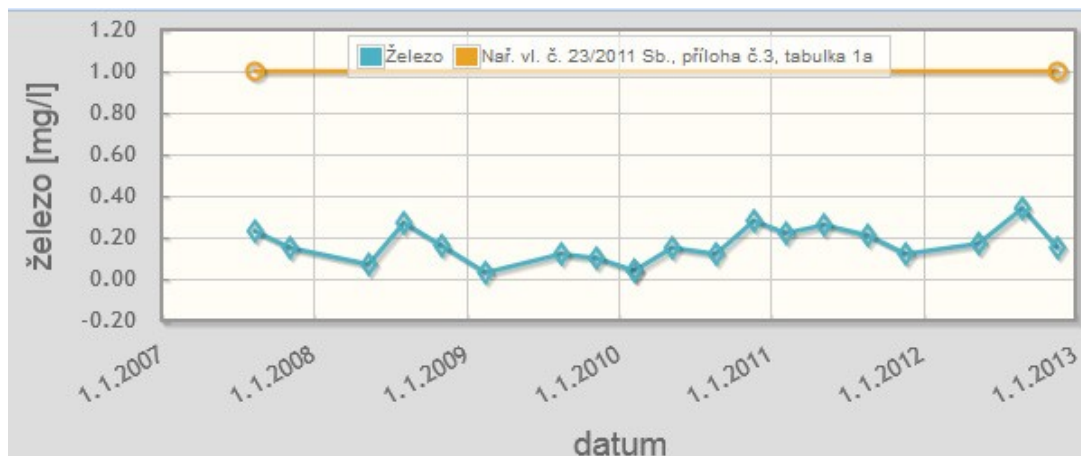
Dostávám se k vytvoření potřebného pole *features*. Naplním jej všemi stanicemi a potřebnými hodnotami k zobrazení metodou *push*. Vše proběhne pomocí cyklu *for*, hlavními parametry jsou souřadnice pro jednoznačné umístění bodu. Následují kritéria mnou dodefinovaná. Potřebuji znát unikátní číslo vrtu dle databáze pro akci kliknutí myši a pro popup okno informace například *Surface elevation*, *Class*, *Účel*, *Využití*, *atd.* (Obrázek 11: Ukázka popup okna). Kombinace anglického a českého jazyka vyplývá z požadavku vedoucího. Zde se mi vyskytuje neočekávané chování při vybrání stanice, její opuštění myši a opětovné označení. Popup okno zůstává otevřeno, již nezmizí. Upravím samotnou knihovnu OpenLayers.js přidáním řádku tam, kde by okno mělo být zavřeno.



Obrázek 11: Ukázka popup okna

V případě výběru konkrétního bodu v mapě kliknutím myši, je vyvolán výběr a v levém sloupci se zobrazí konkrétní informace. Dále musím konkretizovat požadavek výběrem časového období do dvou textových polí. Automaticky se vyplní první a poslední den, měsíc, rok u vybrané stanice. V posledním kroku je třeba vybrat požadovanou veličinu (vápník, teplotu vody, železo, a podobně).

U vybraných položek mapy lze nalézt najednou několik takzvaných sledování. Například se jedná o hodnoty naměřené v jednotlivých výškách vodního sloupce. Na vzniklou situaci reaguji přidáním výběrového menu, kde tyto pozorování vypíšu, a uživatel zvolí vhodnou položku. Poté tlačítkem *Zobrazit údaje* získáme tabulku s výpisem všech navolených údajů.

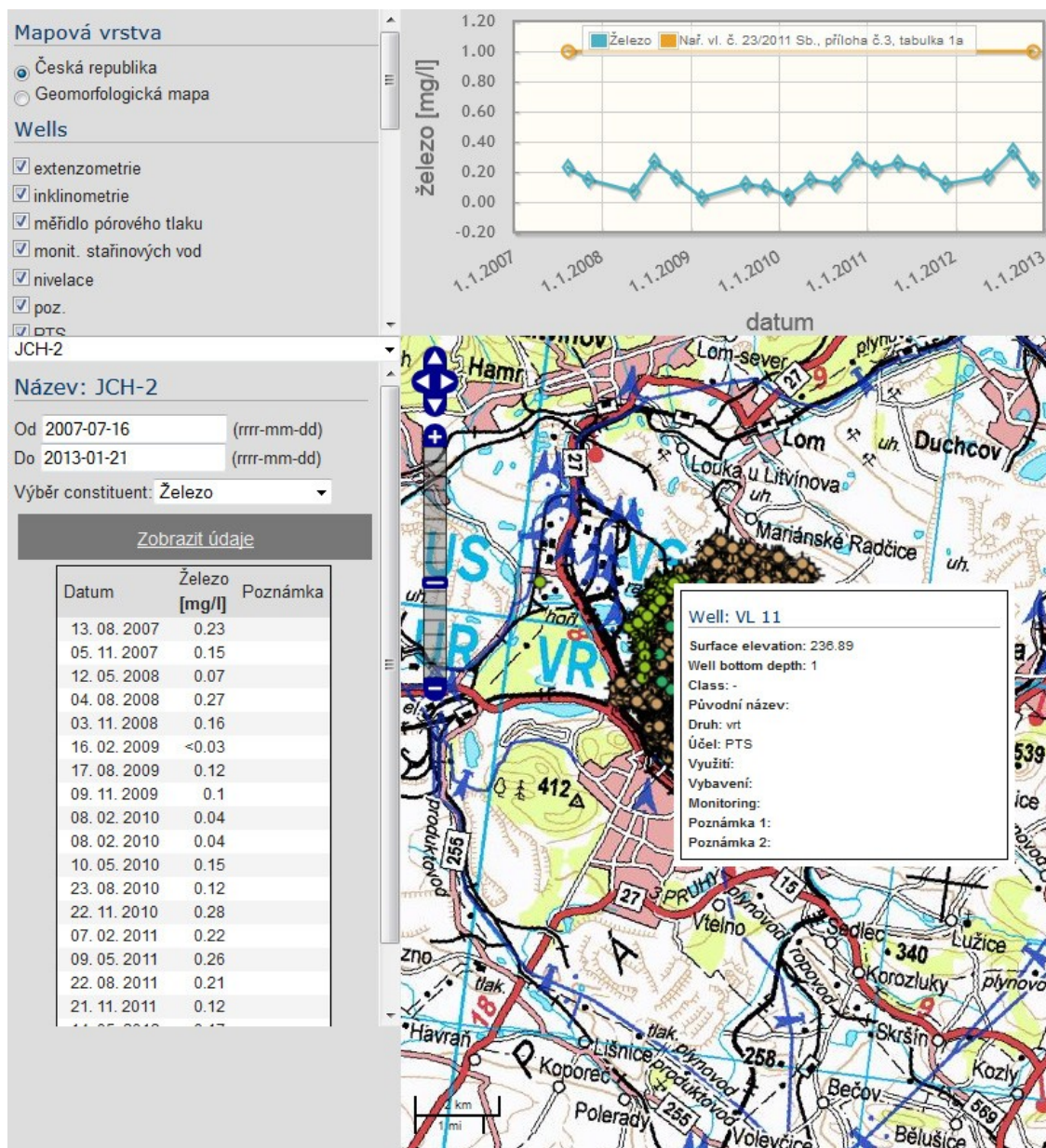


Obrázek 12: Příklad zobrazení grafu

V horní části vykresluji graf (Obrázek 12: Příklad zobrazení grafu) odpovídající hodnotám v tabulce na základě zvolené veličiny a pozorování. Graf také reaguje na pohyb myši. Pohybem nad jednotlivými body získám přesný datum a hodnotu vztahující se k danému datu. Hodnota je zobrazena včetně příslušných jednotek. Oranžová čára v grafu značí daný standard pro danou veličinu.

Stanici můžeme vybrat ve výběrovém menu nacházející se taktéž v levém sloupci. Zde najdeme více vrtů, než vidíme v mapě. Důvodem jejich nezobrazení jsou nevyplněné údaje o poloze v databázi. I přesto chceme sledovat jejich naměřené hodnoty.

Mám k dispozici dva typy souřadnic. První souřadnice obsahují informace o zeměpisné délce a šířce, druhý typ obsahuje záporné souřadnice systému S-JTSK/Krovak East North [13].



Obrázek 13: Konečný vzhled aplikace

## 5.5 Hosting aplikace

Pro bezproblémový chod a dostupnost internetových stránek projektu je třeba umístění na webhostingu či robustnějším webovém serveru. V mém případě se jedná o virtuální server.

Na webovém serveru mám vytvořený nový soubor definující prostor pro svůj projekt. Tento soubor se nazývá virtualhost. Abych mohl nahrát potřebné soubory, potřebuji zřídit FTP přístup do konkrétní složky, odkud budou poté stránky načítány. Vytvořím příslušného uživatele se svými právy ke složce na webovém serveru. V této chvíli mohu celou složku s projektem zkopírovat na server a po zadání adresy se stránky zobrazí.

## 6 Vyhodnocení řešení

### 6.1 Serverové požadavky

Aplikace může fungovat na jakémkoliv vlastním soukromém webovém serveru nebo jí můžeme umístit na placený hosting. Podmínkou je dnes ne tak obvykle vyskytující se podpora pro PostgreSQL databázi. Naopak skriptovací jazyk PHP bývá velice často standardní výbavou placených webhostingů. Výhoda řešení je v jednoduchosti, stačí nakopírovat obsah aplikace pomocí FTP a umístit databázová data. V opačném případě jsme však limitováni diskovým prostorem, především pak prostorem pro databázi. Při dlouhodobém monitoringu data rychle nabývají. S vlastním serverem tyto starosti odpadají. Jako nevýhodu bych zmínil nutnost znalosti serveru po softwarové stránce a jeho správu.

Jedním z kritérií pro pořízení by se dala označit cena. Webové hostingy bývají v řádu desítek korun naproti tomu, vyhovující virtuální server je možné pořídit od stovek korun za měsíc. Klasický dedikovaný server je již zbytečně naddimenzovaný, jeho cena je ještě o řád výše. Pohybuje se zhruba od 2000 Kč za měsíc, dle vybrané konfigurace. Záleží na možnostech nebo náročnosti uživatele. Bude-li se aplikace rozvíjet vysokým tempem, s rostoucím počtem údajů v databázi bude růst i náročnost na diskový prostor a některé běžné hostingy by nemusely dostáčet.

### 6.2 Podpora webových prohlížečů

Aplikace je odladěná pro aktuální verze prohlížečů Microsoft Internet Explorer, Mozilla Firefox a Opera. Ve zmíněných prohlížečích nemá aplikace problém s funkčností. Všechny prvky se vykreslují správně.

Problémy se vyskytují v Google Chrome verze 34, kde program není schopen vykreslit čtrnáct a více skupin vektorů. Jedná se o již známou chybu prohlížeče, byla nahlášena a zahrnuta do jedné z příštích majoritních verzí [2]. Mám proto nainstalovanu také poslední ranou testovací verzi s číslem 36. Aplikaci mám zde otestovanou a je tak pouze otázkou času, kdy bude uvedena a aplikace začne fungovat také pro Google Chrome.

### 6.3 Výběr databázového serveru

Existuje mnoho podobných serverů pro shromažďování dat. Dnes se zde nabízí výběr mezi dnešními nejznámějšími servery, kterými je MSSQL od firmy Microsoft, mnou zvolený PostgreSQL a MySQL. Každý z uvedených má své zastánce či odpůrce. Ohledně řešení od firmy Microsoft je zde jistý fakt, že jej nelze provozovat na jiném systému než je ten od Microsoftu.

### 6.4 Výběr mapového podkladu

Pro výběr samotného mapového podkladu existuje několik knihoven. Nejideálnější a nejlépe popsanou se jevila mapa v kombinaci s knihovnou OpenLayers. Ovšem další možnost je využít knihovnu Leaflet, HSLayers nebo řešení firmy Google, který poskytuje své rozhraní s nutností registrace.



## 7 Shrnutí práce

Práci na nově vytvořené aplikaci bych shrnul v následujících několika odstavcích.

V počátku bylo nutné prozkoumat již hotová řešení, zjistit jejich kladné i záporné vlastnosti. Vhodným způsobem pak implementovat vlastní řešení pro potřeby a požadavky mé bakalářské práce. Hotové aplikace takového rozsahu bývají velmi často zpoplatněny, tudíž existují jako velmi robustní a většina funkcí by mohla zůstat nevyužita.

Pro svůj vývoj tedy potřebuji vhodně implementovaný datový model nad stávajícím modelem, vycházející z EI MARE v databázi Microsoft Access. Tato část se podařila, vznikl tak model pro uchování dat o jednotlivých vrtech, jejich měření a hodnotách k jednotlivým veličinám. Je zde také počítáno s budoucím rozvojem aplikace nad rámec bakalářské práce, proto se v již hotové databázi vyskytují prázdné tabulky a prozatím nejsou využity.

Navržená webová stránka klade důraz na funkčnost a rychlost reakce na uživatelské pokyny. Databázové dotazy jsou optimalizované pro rychlost načítání údajů z databáze, nevznikají zde několikasekundové prodlevy při návštěvníkově interakci s prostředím. Podařilo se mi dosáhnout, pomocí javascriptových knihoven, uživatelsky přívětivé aplikace, kde každé kliknutí myši neznamena znovu načíst celou stránku. Technologií AJAX chování eliminuji a měním pouze vybrané části, kterých se změna nějakým způsobem dotýká – například změna sledovaného vrtu s kliknutím myši do mapy, vykreslení grafu a podobně.

Výsledné webové stránky fungují na mém vlastním virtuálním serveru, kde si mohu nakonfigurovat vše, jak potřebuji. Tudíž má volba se uchýlila tímto směrem a stránky tak fungují i pod vlastním doménovým jménem. V konečném důsledku je však řešení nezávislé a v případě, splnění požadavků na aplikaci bude fungovat také kdekoli jinde.

V průběhu vytváření jsem zjistil nedostatky v prohlížeči Google Chrome, ačkoliv náprava byla přislíbena, je aplikace ochuzena o funkčnost v tomto prohlížeči. Dále bylo nutné upravit samotnou knihovnu OpenLayers, aby popup okna k vrtům bylo možné skrývat při jeho opuštění myší. V kombinaci s událostí kliknutím myši se bohužel jedná o vlastnost než o chybu. Vzhledem k vývoji již třetí verze této knihovny můžeme doufat v úpravu chování či přepracování událostí pro chování myši.



Jak jsem již zmínil, do budoucna je aplikace připravena pro další rozšiřování. Bude tak možno zobrazovat data z doposud nevyužitých tabulek v databázi. Dále počítám v případě potřeby s možností rozšířit mapové podklady.

## Citace

- [1] CENIA. *Národní geoportál INSPIRE* [online]. 2010 [cit. 2014-05-03]. Dostupné z: <http://geoportal.gov.cz/web/guest/home>
- [2] Issue 346621: Rendering/compositing error when stacking many render layers. *Google* [online]. 2014 [cit. 2014-05-04]. Dostupné z: <https://code.google.com/p/chromium/issues/detail?id=346621>
- [3] Data Modeling & Profiling Tool: SQL Power Architect. *SQL Power Software* [online]. 2014 [cit. 2014-05-06]. Dostupné z: <http://www.sqlpower.ca/page/architect>
- [4] *Open Geospatial Consortium* [online]. 2014 [cit. 2014-05-06]. Dostupné z: <http://www.opengeospatial.org/>
- [5] Open Geospatial Consortium: Standards. *Open Geospatial Consortium* [online]. 2014 [cit. 2014-05-06]. Dostupné z: <http://www.opengeospatial.org/standards>
- [6] Geography Markup Language. *Open Geospatial Consortium* [online]. 2014 [cit. 2014-05-10]. Dostupné z: <http://www.opengeospatial.org/standards/gml>
- [7] JODAS, Tomáš. *Webová prezentace prostorových dat*. Liberec, 2013. Bakalářský projekt. Technická univerzita v Liberci. Vedoucí práce Mgr. Kamil Nešetřil.
- [8] GEOWISE LTD. *Interactive mapping software* [online]. 2013 [cit. 2014-05-11]. Dostupné z: <http://www.instantatlas.com/>
- [9] SATSILK. *Statsilk* [online]. 2014 [cit. 2014-05-11]. Dostupné z: <http://www.statsilk.com/>
- [10] DE LA BEAUJARDIERE, Jeff. OPEN GEOSPATIAL CONSORTIUM. *OpenGIS® Web Map Server Implementation Specification*. 2006. Dostupné z: [http://portal.opengeospatial.org/files/?artifact\\_id=14416](http://portal.opengeospatial.org/files/?artifact_id=14416)
- [11] MASÓ, Joan, Keith POMAKIS a Núria JULIÀ. OPEN GEOSPATIAL CONSORTIUM. *OpenGIS® Web Map Tile Service Implementation Standard*. 2010. Dostupné z: [http://portal.opengeospatial.org/files/?artifact\\_id=35326](http://portal.opengeospatial.org/files/?artifact_id=35326)

- [12] VRETANOS, Panagiotis A. OPEN GEOSPATIAL CONSORTIUM. *OpenGIS Web Feature Service 2.0 Interface Standard*. 2010. Dostupné z: [http://portal.opengeospatial.org/files/?artifact\\_id=39967](http://portal.opengeospatial.org/files/?artifact_id=39967)
- [13] Souřadnicové systémy. ČÚZK. *Geoportál ČÚZK* [online]. 2010 [cit. 2014-05-13]. Dostupné z: [http://geoportal.cuzk.cz/\(S\(lbthf0ik1dqs35551autuc45\)\)/Default.aspx?mode=TextMeta&side=sit.trans&text=souradsystemy](http://geoportal.cuzk.cz/(S(lbthf0ik1dqs35551autuc45))/Default.aspx?mode=TextMeta&side=sit.trans&text=souradsystemy)
- [14] OPEN SOURCE GEOSPATIAL FOUNDATION. *OpenLayers* [online]. 2014 [cit. 2014-05-14]. Dostupné z: <http://www.openlayers.org/>
- [15] *HSLayers* [online]. 2012 [cit. 2014-05-14]. Dostupné z: <http://www.hslayers.org/>
- [16] *Leaflet: An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps* [online]. 2010 [cit. 2014-05-14]. Dostupné z: <http://www.leafletjs.com/>
- [17] SENCHA INC. *Sencha Ext JS: JavaScript Framework for Rich Desktop Apps* [online]. 2014 [cit. 2014-05-14]. Dostupné z: <http://www.sencha.com/products/extjs/>
- [18] GOOGLE INC. *Google developers: Google Maps API* [online]. 2014 [cit. 2014-05-14]. Dostupné z: <https://developers.google.com/maps/?hl=cs>
- [19] *PhpPgAdmin* [online]. 2013 [cit. 2014-05-15]. Dostupné z: <http://phppgadmin.sourceforge.net/doku.php>